

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application Serial No. 09/608,028
Filing Date June 30, 2000
Inventor Kartik N. Raghavan et al.
Group Art Unit 2136
Examiner COLIN, CARL G.
Attorney's Docket No. MS1-0498US
Confirmation No. 9044
Title: A System and Related Methods for Automatically Configuring a Computing System

To: The Honorable Commissioner for Patents
 Mail Stop Appeal Brief- Patents
 PO Box 1450
 Alexandria, Virginia 22313-1450

From: Beatrice L. Koempel-Thomas (Tel. 509-324-9256 x259; Fax 509-323-8979)
 Customer No. 22801

BRIEF OF APPELLANT

The Applicant has filed a timely Notice of Appeal from the action of the Examiner in finally rejecting all of the claims that were considered in this application. This Brief is being filed under the provisions of 37 C.F.R. § 41.37. The Filing Fee, as set forth in 37 C.F.R. § 41.20(b)(2) and the appropriate forms accompany this Brief for payment of any additional fees (e.g., for a two-month extension).

TABLE OF CONTENTS

I.	Real Party in Interest	Page 3
II.	Related Appeals and Interferences	Page 4
III.	Status of Claims	Page 5
IV.	Status of Amendments	Page 9
V.	Summary of the Claimed Subject Matter	Page 10
VI.	Grounds of Rejection to be Reviewed on Appeal	Page 22
VII.	Argument	Page 24
VIII.	Appendix of Appealed Claims	Page 51
IX.	Appendix of Evidence	Page 63
X.	Appendix of Related Appeals and Interferences	Page 64

I. REAL PARTY IN INTEREST

The real party in interest is Microsoft Corporation, by way of assignment from Raghavan et al., who is the named inventive entity and is captioned in the present brief.

II. RELATED APPEALS AND INTERFERENCES

None.

III. STATUS OF CLAIMS

Claims 1-36 are rejected. The rejections of claims 1-36 are the subject of this appeal.

The history of the claims is as follows:

- a. Claims 1-36 were originally filed.
- b. In an Office Action mailed January 29, 2004, claims 1-26 were rejected under 35 U.S.C. 102(e) as being anticipated by Liming, U.S. Patent Application Pub. 2002/0055924; claims 32-36 were rejected under §102(e) as being anticipated by Hansen U.S. Patent No. 5,838,907; and claims 27-31 were rejected under 35 U.S.C. 103(a) as being unpatentable over Liming in view of Hansen.
- c. An Office Action Response was filed on April 29, 2004 where Applicant submitted a §1.131 affidavit and evidence removing Liming as a reference and amended claims 15, 20 and 21.
- d. In an Office Action mailed September 13, 2004, claims 1-26 and 32-36 were rejected under 35 U.S.C. 102(e) as being anticipated by Li et al. U.S. Patent No. 6,012,088; and claims 27-31 were rejected under 35 U.S.C. 103(a) as being unpatentable over Li et al. in view of Chrabaszcz U.S. Patent No. 6,212,585.
- e. An Office Action Response was filed on November 19, 2004 where Applicant amended claims 27 and 32.

- f. In a Final Office Action mailed March 2, 2005, claims 1-26 and 32-36 were rejected under 35 U.S.C. 102(e) as being anticipated by Li et al. U.S. Patent No. 6,012,088; and claims 27-31 were rejected under 35 U.S.C. 103(a) as being unpatentable over Li et al. in view of Chrabaszcz U.S. Patent No. 6,212,585.
- g. An Examiner Interview was held on May 5, 2005.
- h. An Interview Summary was mailed on May 9, 2005.
- i. An Office Action Response was filed on July 1, 2005 where Applicant amended claims 14 and 16.
- j. An Advisory Action was mailed on July 28, 2005.
- k. A Request for Continued Examination was filed with an Office Action Response on August 25, 2005 where Applicant amended claims 1, 4-6, 8, 13-16, 18, 21-22, 26-27, 29, 32, 34, and 36.
- l. In an Office Action mailed November 14, 2005, claims 1-36 were rejected under 35 U.S.C. 103(a) as being unpatentable over Stupek, Jr. et al. U.S. Patent No. 5,809,287 in view of Li et al. U.S. Patent No. 6,012,088.
- m. An Examiner Interview was held on January 25, 2006.
- n. An Office Action Response was filed on February 14, 2006 where Applicant amended claims 1, 13, 17, 21, 24, 26-27, 32 and 34.
- o. An Interview Summary was mailed on February 23, 2006.

- p. In a Final Office Action mailed May 2, 2006, claims 1-36 were rejected under 35 U.S.C. 103(a) as being unpatentable over Stupek, Jr. et al. U.S. Patent No. 5,809,287 in view of Li et al. U.S. Patent No. 6,012,088.
- q. An Office Action Response was filed on June 30, 2006 where Applicant amended claims 1, 2, 13, 21, 27 and 32.
- r. An Advisory Action was mailed on July 19, 2006.
- s. A Request for Continued Examination was filed with an Office Action Response on August 2, 2006 where Applicant amended claims 1, 2, 13, 21, 27 and 32.
- t. In an Office Action mailed October 13, 2006, claims 1-36 were rejected under 35 U.S.C. 102(e) as being anticipated by Cheng et al. U.S. Patent No. 6,151,643.
- u. An Office Action Response was filed on January 12, 2007 where Applicant amended claims 1 and 21.
- v. In a Final Office Action mailed April 12, 2007, claims 1-36 were rejected under 35 U.S.C. 102(e) as being anticipated by Cheng et al. U.S. Patent No. 6,151,643.
- w. An Examiner Interview was held on June 27, 2007.
- x. An Interview Summary was mailed on July 10, 2007.

- y. An Office Action Response was filed on August 10, 2007 where Applicant amended claims 1, 2, 4, 6, 13, 21, 27, 28, 32 and 36.
- z. An Advisory Action was mailed on August 23, 2007.
- aa. Appellant filed a Notice of Appeal on September 10, 2007.

IV. STATUS OF AMENDMENTS

Amendments submitted in the Office Action Response filed January 12, 2007 have been entered.

Amendments submitted in the Office Action Response after Final Rejection filed August 10, 2007 have not been entered.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The initial setup and/or reconfiguration of computing devices is an area that had not kept pace with other advances in computing technology. Prior to this invention the state-of-the-art of installing and configuring a computing system involved a collection of separate and disjointed processes to locate and retrieve the information necessary from disparate sources often located on different networks and requiring the manual interaction of technically competent technicians to install and properly configure the operating system, Internet settings and applications on the computing system. Specifically, installation configuration information had to be gathered from a number of separate places.

Even once this initial set-up had been completed, additional time was required of the end-user to personalize the settings. Installation of the operating system (OS) typically required a password for the computer, the name of the computer, and the type of network to which the computer would belong. Manual, user-interaction and a prior understanding of which OS components and features of the operating system to enable was required before the personalization and licensing options of the operating system were completed. Configuration and personalization of the Internet browser and other applications required similar manual interaction. The disparate nature/sources of the information had heretofore stifled attempts to automate the configuration/reconfiguration process. *See Application, Page 1, paragraph 2 to Page 2, paragraph 2.*

Accordingly, this system and related methods for automatically configuring a computing device was invented. One of the claimed aspects is a method of configuring a computing system that comprises at a server, receiving an access request from the computing system, wherein an identifier associated with a user and/or the computing system is provided, and downloading one or more image files to automatically install/update one or more of an operating system, application(s), and personalized configuration setting information. Thus, the claimed invention provides a mechanism of automatically installing and configuring an operating system, applications and personalization settings of a computing system. *See Application, Page 2, paragraph 3.*

For example, Fig. 7 is a flowchart illustrating an exemplary method of automatically configuring/reconfiguring a computing system. A configuration agent at the server receives a unique identifier associated with a computing system or computing system user. More specifically, control logic receives an identifier via a communicatively coupled network via a network interface. Notably, the identifier need not be issued by the computing system itself, but may be sent by another computing system or communications device. In response to receiving the identifier, the control logic invokes the identification facilities of identification and characterization function to access a particular data structure and equate the received identifier with a known user, user group, or computing system configuration.

The configuration agent assesses the current resources of the identified computing system by invoking the characterization facilities of its identification and characterization

function to assess the system resources of the computing system. The identification and characterization function communicates with the computing system via the network interface and the configuration interface. The identification and characterization function interrogates one or more system resources (e.g., a processor, memory systems, storage devices, etc.) to develop an assessment of the current system resources of the computing system. Based on the review of system resources, the identification and characterization function provides the control logic with a report of the current system resources. The control logic identifies resources which require installation, configuration and/or reconfiguration on the computing system based on the report by comparing the report against the authorized and available resources associated with the computing system in the data structure. Based on the identified resource needs, the control logic selectively accesses one or more data stores to collect the identified resources. In one implementation, the identified resources are compressed and stored as a self-extracting image file that is downloaded to the computing system. Once the self-extracting image file is completely downloaded, it is automatically installed on the computing system. Thus, a computer system is configured with all authorized and available resources (including configuration and personalization settings) automatically. *See Application, Page 18, paragraph 2 to Page 21, paragraph 4.*

Independent Claim 1 recites a method comprising:

- maintaining, on a server for each of a plurality of computing systems, data specifying which resources are authorized for the computing system (e.g., reference numbers 12 and 16, Fig. 1; page 4, paragraph 2);
- receiving by a configuration agent at the server an identifier associated with a computing system and/or computing system user, the configuration agent (e.g., reference number 14, Figs. 1 and 4, page 4, paragraph 2, page 5, paragraph 2, and page 12, paragraph 2; and reference number 702, Fig. 7, page 19 paragraph 1):
 - obtaining, from the server, data that specifies authorized resources corresponding to the received identifier (e.g., reference numbers 402 and 408, Fig. 4, and reference numbers 502 and 504, Fig. 5; page 17, paragraph 2; and reference number 704, Fig. 7; page 19, paragraph 2);
 - interrogating the computing system to produce an assessment indicating existing hardware and/or software resources available on the computing system (e.g., reference numbers 402 and 406, Fig. 4, page 15, paragraph 1; and reference number 706, Fig. 7; page 19, paragraph 3);
 - comparing the authorized resources with the assessment to identify one or more resources authorized but not installed on the computing system (e.g., reference numbers 402 and 406, Fig. 4, and reference numbers 506-510, Fig. 5; page 17, paragraph 2; and reference number 708, Fig. 7; page 20,

paragraph 1); and

- automatically modifying the computing system resources by installing the one or more identified resources (e.g., reference number 402, Fig. 4; and reference number 710, Fig. 7; page 20, paragraph 2).

Dependent claim 2 recites a method according to claim 1, wherein the computing system is provided to the user without the authorized resources being preinstalled (e.g., reference number 18A, Fig. 1; and page 11, paragraph 1).

Independent Claim 13 recites a server comprising:

- a storage device to maintain a profile of personal resources specifying, for each of a plurality of computing systems, which resources are authorized for the computing system (e.g., reference numbers 12 and 16, Fig. 1; page 4, paragraph 2; reference numbers 402 and 408, Fig. 4, and reference numbers 502 and 504, Fig. 5; page 17, paragraph 2; and reference number 704, Fig. 7; page 19, paragraph 2); and
- a configuration agent, coupled to the storage device (e.g., reference number 14, Figs. 1 and 4, page 4, paragraph 2, page 5, paragraph 2, and page 12, paragraph 2), to:
 - receive an identifier associated with a computing system and/or computing system user (e.g., reference number 14, Figs. 1 and 4, page 4, paragraph 2, page 5, paragraph 2, and page 12, paragraph 2; and reference number 702, Fig. 7, page 19 paragraph 1);
 - generate an assessment of the current resources of the computing system (e.g., reference numbers 402 and 406, Fig. 4, page 15, paragraph 1; and reference number 706, Fig. 7; page 19, paragraph 3);

- identify, by comparing the assessment with the authorized resources, one or more of the authorized resources which are missing from a computing system (e.g., reference numbers 402 and 406, Fig. 4, page 15, paragraph 1; and reference number 706, Fig. 7; page 19, paragraph 3); and
- automatically configure resources of the computing system to include the identified resources (e.g., reference number 402, Fig. 4; and reference number 710, Fig. 7; page 20, paragraph 2).

Independent claim 21 recites a storage medium comprising a plurality of executable instructions including at least a subset of which that, when executed, implement a configuration agent at a server to:

- maintain, for each of a plurality of computing systems, data specifying authorized resources for the computing system (e.g., reference numbers 12 and 16, Fig. 1; page 4, paragraph 2; reference numbers 402 and 408, Fig. 4, and reference numbers 502 and 504, Fig. 5; page 17, paragraph 2);
- conduct an assessment of computing system resources upon receipt of an identifier associated with the computing system and/or computing system (e.g., reference number 14, Figs. 1 and 4, page 4, paragraph 2, page 5, paragraph 2, and page 12, paragraph 2; and reference numbers 702, 704, and 706 Fig. 7, page 19 paragraphs 1-3);
- identify, by comparing the assessment with corresponding data specifying authorized resources, one or more of the authorized resources which are missing from the computing system (e.g., reference numbers 402 and 406, Fig. 4, page 15, paragraph 1; and reference number 706, Fig. 7; page 19, paragraph 3); and
- automatically download and install on the computing system the missing authorized resources (e.g., ., reference number 402, Fig. 4; and reference number 710, Fig. 7; page 20, paragraph 2).

Independent Claim 27 recites a computing system comprising:

- a storage device having stored thereon a plurality of executable instructions (e.g., reference numbers 18A and 18B, Fig. 1; page 10, paragraph 2-page 11, paragraph 1; reference numbers 18 and 304, Fig. 3);
- a network interface, communicatively coupling the computing system to a network; (e.g., reference numbers 18A, 18B, 20, and 22 Fig. 1; page 4, paragraph 2-page 5 paragraph 1); and
- a controller, coupled to the storage device and the network interface, to execute at least a subset of the plurality of executable instructions to make an assessment of current hardware and/or software resources of the computing system, and to implement a basic input/output system (BIOS) to issue a configuration request to the network via the network interface, the configuration request based on the assessment and including an identifier associated with the computing system, wherein the configuration request is configured to cause a recipient of the request to (e.g., reference numbers 18 and 302, Fig. 3, page 10, paragraph 2-page 11, paragraph 1):

- reference the identifier to access corresponding data specifying authorized resources associated by the identifier with the computing system (e.g., reference number 14, Figs. 1 and 4, page 4, paragraph 2, page 5, paragraph 2, and page 11, paragraph 2-page 12, paragraph 2; and reference number 702 and 704, Fig. 7, page 19 paragraph 1);
- compare the assessment to the authorized resources to determine one or more of the authorized resources missing from the computing system (e.g., reference numbers 402 and 406, Fig. 4, page 15, paragraph 1; and reference number 708, Fig. 7; page 19, paragraph 3-page 20, paragraph 1) ; and
- provide the missing authorized resources to the computing system via the network (e.g., reference number 710, Fig. 7; page 20, paragraph 2).

Dependent claim 28 recites a computing system according to claim 27, wherein the computing system is an unconfigured computing system (e.g., page 5 paragraph 2; and page 11, paragraph 1).

Independent Claim 32 recites a method comprising:

- issuing a configuration request from a computing system, wherein the configuration request includes an identifier associated with the computing system and/or computing system user and is configured to cause a recipient of the request to (e.g., reference number 18, Fig. 3, page 10, paragraph 2-page 11, paragraph 3; and reference number 702, Fig. 7, page 19 paragraph 1);
 - generate an assessment of the current computing system resources of the computing system (e.g., reference number 706, Fig. 7; page 19, paragraph 3);
 - reference the identifier to access data specifying authorized computing system resources associated by the identifier with the computing system (e.g., reference number 704, Fig. 7; page 19, paragraph 2); and
 - compare the assessment to the authorized computing system resources to determine one or more of the authorized computing system resources missing from the computing system (e.g., reference number 708, Fig. 7; page 20, paragraph 1); and

- receiving a response to the configuration request at the computing system, the response including the one or more computing system resources missing from the computing system, wherein the one or more computing system resources are automatically installed and configured on the computing system (e.g., ., reference number 402, Fig. 4; reference number 710, Fig. 7; page 20, paragraphs 2-3; and page 21, paragraphs 2-3).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Appellant respectfully requests that the Board review the grounds, as stated by the Examiner, for rejection of all 36 claims (1-36) in the instant application as being anticipated by U.S. Patent No. 6,151,643 to Cheng et al. (hereinafter "Cheng"), under 35 U.S.C. 102(e).

Two grounds of rejection are disputed as applied to the independent claims:

The first issue in dispute is whether Cheng discloses the "configuration agent" of the claims; the Examiner asserts that Cheng discloses the "configuration agent" of the claims and thus anticipates the subject matter of all thirty-six claims (1-36). Appellant disagrees. Further, the evidence is insufficient to support the Examiner's finding. Appellant requests a decision on this issue. In particular, Appellant requests review of the grounds of rejection, based on specific evidence and argument supplied by the Examiner, which are of record in the Office Action of (October 13, 2006) and the Final Office Action of (April 12, 2007).

The second issue in dispute is whether Cheng discloses the *authorized resources* of the claims; the Examiner asserts that Cheng discloses the *authorized resources* of the claims and thus anticipates the subject matter of all thirty-six claims (1-36). Appellant disagrees. Further, the evidence is insufficient to support the Examiner's finding. Appellant requests a decision on this issue. In particular, Appellant requests review of the grounds of rejection, based on specific evidence and argument supplied by the Examiner, also of record in the Office Action of (October 13, 2006) and the Final Office Action of (April 12, 2007).

A third ground of rejection is disputed as applied to the dependent claims:

The third issue pertains to whether the Application as filed supports and the Cheng reference discloses the subject matter of the dependent claims 2 and 28, for example, “without the authorized resources being preinstalled” of claim 2 and “the computing system is an unconfigured computing system” of claim 28. Appellant requests a decision on the issue of these claims as well.

VII. ARGUMENT

Appellant disputes three grounds of rejection. Appellant submits that the Office erred in rejecting all 36 claims (1-36) under 35 U.S.C. §102(e) because the evidence does not support rejection of each and every element set forth in the claims. Appellant requests that the Board review the Examiner's grounds for rejection of all 36 claims (1-36) in the instant application as being anticipated by U.S. Patent No. 6,151,643 to Cheng et al. (hereinafter "Cheng"), under 35 U.S.C. 102(e).

Specific Errors

(a) Independent Claims. Appellant submits that the Office erred in rejecting claims 1, 13, 21, 27, and 32 under 35 U.S.C. §102(e) because Cheng does not disclose each and every element set forth in the independent claims. For example, Cheng does not disclose the *configuration agent* and *authorized resources* recited in these independent claims. Appellant addresses arguments to the independent claims representing all 36 claims (1-36) as a group, with claim 1 being exemplary.

(b) Dependent Claims. Appellant submits that the Office erred in rejecting all of the dependent claims under 35 U.S.C. §102(e) because in addition to Cheng not disclosing each and every element set forth in the independent claims, Cheng does not disclose the subject matter of dependent claims 2 and 28 "without the authorized resources being preinstalled" or "an unconfigured computing system" (e.g., a new computer, etc.), respectively.

Standards

All claims are rejected under 35 U.S.C. §102(e). According to MPEP §2131 “a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference”.

The examiner has the initial burden of establishing a prima facie case of anticipation by pointing out where all of the claim limitations appear in a single reference. See In re Spada, 911 F.2d 705, 708, 15 USPQ2d 1655, 1657 (Fed. Cir. 1990); In re King, 801 F.2d 1324, 1327, 231 USPQ 136, 138-39 (Fed. Cir. 1986). In order for a claimed invention to be anticipated under 35 U.S.C. § 102, all of the elements of the claim must be found in one reference. See Scripps Clinic & Research Found. v. Genentech Inc., 927 F.2d 1565, 1576, 18 USPQ2d 1001, 1010 (Fed. Cir. 1991).

A reference must enable one of skill in the art to make the anticipating subject matter. Scripps Clinic & Research Fdn., 927 F.2d at 1576. If a cited reference is not enabling of anticipating subject matter, that subject matter cannot be used as a basis for a §102 rejection.

(a) Rejection of Independent Claims

Claims 1-36 as a Group represented by Claim 1:

For purposes of expediency, Appellant discusses the errors in the rejection of the independent claims by presenting claim 1 as an example. Claim 1 recites (emphasis of the authorized resources and configuration agent in bold):

A method comprising:

*maintaining, on a server for each of a plurality of computing systems, data specifying **which resources are authorized for the computing system**;*

*receiving by a **configuration agent** at the server an identifier associated with a computing system and/or computing system user, **the configuration agent**:*

***obtaining**, from the server, data that specifies **authorized resources** corresponding to the received identifier;*

***interrogating** the computing system to produce an assessment indicating existing hardware and/or software resources available on the computing system;*

***comparing** the authorized resources with the assessment to identify one or more resources authorized but not installed on the computing system; and*

***automatically modifying** the computing system resources by installing the one or more identified resources.*

FIRST GROUND OF REJECTION (addressed in detail, below): Thus, per claim 1: (i) a configuration agent at a server receives an identifier associated with a computing system and/or computing system user; (ii) the configuration agent obtains data that specifies authorized resources corresponding to the received identifier from the server; (iii) the configuration agent interrogates the computing system to produce an assessment indicating existing hardware and/or software resources available on the computing system; (iv) the configuration agent compares the authorized resources with the assessment to identify one or more resources authorized but not installed on the computing system; and (v) the configuration agent automatically modifies the computing system resources by installing the one or more identified resources. Thus, the configuration agent at the server receives the indication and performs the tasks of obtaining, interrogating, comparing, and automatically modifying. Appellant respectfully submits that this meaning was previously discussed with the Office as evidenced by the record (Response of 1/12/07).

To properly reject claim 1 as being anticipated, Cheng must disclose this configuration agent, i.e., a configuration agent that performs the tasks of obtaining, interrogating, comparing, and automatically modifying as claimed. Again, Appellant maintains that Cheng does not disclose such a configuration agent.

SECOND GROUND OF REJECTION (addressed in detail, below): Further, per claim 1: (vi) data is maintained on the server for each of a plurality of computing systems specifying which resources are authorized for the computing system. Thus, the data

maintained on the server specifies which resources are authorized for each computing system. Appellant respectfully submits that this meaning was previously discussed with the Office as evidenced by the record (Response of 1/12/07).

To properly reject claim 1 as being anticipated, Cheng must disclose such data maintained on the server showing authorized resources, i.e., "data specifying which resources are authorized for the computing system" as claimed. Again, Appellant maintains that Cheng does not disclose this.

Stated Grounds of Rejection of Claim 1

In the Final Office Action (4/12/07, pp. 5-6), the Office stated the following grounds for rejection of claim 1:

As per claim 1, Cheng et al discloses a method comprising: maintaining, on a server for each of a plurality of computing systems, database including specific data, resources, profile, activity log, payment information, etc. associated with each user computing system that meets the recitation of data specifying which resources are authorized for the computing system (see column 17, lines 1-21 and figure 7); in column 6, lines 44-50 and column 20, lines 33-60 it is disclosed that authentication information is also stored associated with update software and products which meets the recitation of data specifying which resources are authorized for the computing system; column 7, lines 11-39 describes that users must be registered to received authorized resources. Cheng et al discloses receiving by a configuration agent (700, figure 7) at the server an identifier (i.e. user ID, password, unique registration-number, e-mail address, or authentication information) associated with a computing system and/or computing system user

(column 7, lines 10-40; column 5, lines 18-21; and column 23, lines 41-49); and discloses that the authentication ensures that only users who are authorized can obtain updates for software products (see column 7, lines 40-45). Cheng et al discloses a payment module obtaining user information such as fee, usage, count, connection time (data) from the server database that specifies authorized resources (software updates) corresponding to the received user login information (see column 17, lines 2-15 and lines 40-47) that meets the recitation of the configuration agent obtaining from the server data that specifies authorized resources corresponding to the received identifier. Cheng et al discloses the configuration agent interrogating the computing system to produce an assessment indicating assessing existing hardware and/or software computing system resources available on the computing system (see column 18, lines 42-58 and figures 17a-d); the configuration agent comparing the authorized resources with the assessment to identify one or more resources authorized but not installed on the computing system (see column 14, lines 45-64); and the configuration agent automatically modifying the computing system resources by installing the one or more identified resources (see column 15, lines 16-24 and column 4, lines 12-23 and column 27, lines 24-26).

The Office's argument relies on the addressable memory (Fig. 7, item 700) disclosed in Cheng. This evidence is identified and discussed below.

FIRST GROUND OF REJECTION:

Cheng: Evidence of "addressable memory 700"

Cheng is entitled "Automatic **Updating** of Diverse Software Products on Multiple Client Computer Systems by **Downloading** Scanning Application to **Client Computer** and **Generating Software List on Client Computer**" (emphasis added) and includes figures 1-17d. Briefly, Fig. 7 shows the software architecture of Cheng's service provider computer

where “addressable memory” is labeled “700,” further, the text refers to item 700 as “addressable memory” (*see, e.g.*, col. 9, line 66).

Appellant presents Fig. 7 of Cheng, below:

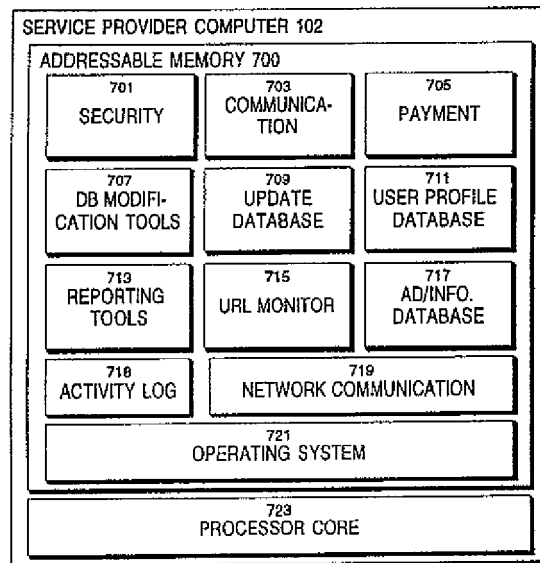


FIG. 7

As indicated in Cheng, Fig. 7 shows “the service provider computer 102. In terms of hardware architecture, the service provider computer 102 is conventional server type computer, preferably supporting a relatively large number of multiple clients simultaneously for requests for data and other processing operations. The service provider computer 102 includes one or more conventional processors in a processor core 723, and a suitable amount of **addressable memory 700**, preferably on the order of 18-64 Mb” (*see* col. 9, lines 56-66, emphasis added).

Specific Evidence Relied on by the Office as Grounds for Rejection

As discussed below, the Office relied on Fig. 7 of Cheng to reject independent claim 1. As indicated in Cheng, Fig. 7 shows “the service provider computer 102 [which] includes . . . a suitable amount of addressable memory 700 . . . the service provider computer 102 includes a number of executable components and database structures useful for managing the software update interactions with the client computer 101 and the software vendor computers 103. These components include a security module 701, a communications module 703, a payment module 705, database modification tools 707, a update database 709, a user profile database 711, a reporting tools module 713, a URL monitor module 715, an advertising/information database 717, and an activity log 718” (*see* col. 9, lines 56-66 and col. 10, lines 13-24).

Based on this evidence, item 700 is not a configuration agent having the functionality claimed. That is, the functionality of the configuration agent in claim 1: “a configuration agent at the server [receiving] an identifier associated with a computing system and/or computing system user, . . . obtaining from the server, data that specifies authorized resources corresponding to the received identifier; interrogating the computing system to produce an assessment indicating existing hardware and/or software resources available on the computing system; comparing the authorized resources with the assessment to identify one or more resources authorized but not installed on the computing system; and automatically modifying the computing system resources by installing the one or more identified

resources.” Rather, Cheng’s item 700 is merely a conventional addressable memory of a Cheng’s service provider computer. Appellant submits that this interpretation is in-line with the purpose of Cheng that “[t]he client computer determines that [sic] software products stored thereon, and using this information, determines from the database, which products have updates available, based on product name and release information for the installed products” (Abstract).

Appellant adds that, under the standard for anticipation, if Cheng does not anticipate each feature of claim 1, then it cannot anticipate each feature of the dependent claims. While not given as the specific grounds for rejection of all thirty-six claims, Appellant adds that Cheng provides insufficient evidence to support a finding that “addressable memory 700” performs the functionality of the method claimed in claim 1. Cheng states: “[R]egistered users are authenticated 203 by the service provider computer 102,” (column 7, lines 40-41).

The service provider computer maintains a database of its users.

This database may be the user profile database 711 . . . the payment module 705 tracks the user’s usage of the service, for example, total the connection time, and maintains a count of the number of software updates downloaded

Cheng, column 17, lines 2-15.

The client application 104 requests, for a software product to be updated, a transaction permission from the service provider computer 102. The payment module 705 determines from the update database 705 a specific fee . . . and returns this information . . . to the client application 104

Cheng, column 17, lines 40-47.

As evidenced by the above excerpts, Cheng's *users* are being authorized by authentication. Appellant submits that this does not amount to disclosure of the recited **configuration agent at the server** "obtaining, from the server, data that specifies **authorized resources** corresponding to the received identifier" of claim 1.

In addition, Cheng states:

Registering a software product has the goal of specifying sufficient information to identify a product and its version if the product has been installed on a given client computer 101. Fig. 17 illustrates a form for registering a software product into the update database 709 for the first time

Cheng, column 18, lines 42-46.

Cheng illustrates example screens of the registration form (Figs. 17a-17d), providing evidence that Cheng's system *interrogates users* to register a software product. Appellant further submits that this does not amount to disclosure of the recited **configuration agent at the server** "interrogating the computing system to produce an assessment indicating existing hardware and/or software resources available on the computing system" of claim 1.

Furthermore, Cheng states:

The analyze process is performed by the system analyzer 907 module of the client application 104 [at the client computer 101.] [T]he process of the system analyzer 907 for analyzing 204 the client computer 101 to determine the list of installed software products[:] first synchroniz[ing] 1001 the method table 801 and the product locator table 803 in the client computer 101 with the current versions held by the service provider computer 102

Cheng, column 13, lines 55-67; *see also* Figures 2 and 9.

[A] list 1013 of the installed software products on the client computer 101[is obtained by the system analyzer 907 from the product locator table 803]

Cheng, column 14, lines 38-40; *see also* Figures 9 and 10.

For each installed product (1002) the system analyzer 907 [at the client computer 101] queries the service provider computer 102 to resolve 1004 the name 815 and release number 818 of the product and determine if there [sic] a current update 821 for the product . . . by passing in the entire list as name, value pairs, or individually quarrying [sic] the service provider computer 102. In either cases, [sic] the service provider computer 102 determines if there is an applicable update for a software product by comparing the product name 815 and release information 818 to the product table 805

Cheng, column 14, lines 45-54; *see also* Figures 8, 9, and 10.

The above excerpts providing evidence that while Cheng's system performs some comparison at the service provider computer, determining the list of installed software products takes place at the client computer and Cheng's evidence is limited to products installed on the client computer. Appellant further submits that this does not amount to disclosure of the recited **configuration agent at the server "comparing the authorized resources with the assessment to identify one or more resources authorized but not installed on the computing system"** of claim 1.

SECOND GROUND OF REJECTION:

Cheng: Evidence of “data specifying which resources are authorized for the computing system”

As discussed below, the Office relied on several additional portions of Cheng to reject independent claim 1. As indicated in Cheng, Fig. 1 shows “a system for providing software updates” (*see* col. 5, lines 36-37). Cheng is directed to:

a system for updating diverse software products on user's computers . . . [i]n system 100, there are a plurality of client computers 101 communicatively coupled by a network 106 to a service provider computer 102. A number of software vendor computers 103 are also communicatively coupled over the network 106 to the service provider computer 102. . . . Each client computer 101 is operated by an end user, and typically has a number of software products installed thereon, such as applications, drivers, utilities and the like. In accordance with [Cheng's] invention, the client computers 101 includes a client application 104 that communicates with the service provider computer 102 to obtain software updates of software products installed on the client computer 101.

Cheng, col. 6, lines 12-28.

Cheng states:

The payment module 705 handles payment by the end user to the service provider for the service of providing software updates [for which] the service provider computer 102 maintains a database of its users . . . [h]owever charged, the payment module tracks the user's usage of the service . . . and maintains a count of the number of software updates downloaded

Cheng, column 17, lines 1-21.

[T]he software vendor computers 103 provide mechanisms for controlling distribution and payment of software updates, such

as credit card payment front ends, code authentication and verification subsystems . . . payment mechanisms may be implemented in compliance with various credit card . . . systems [and] authentication and verification may be implemented using conventional encryption techniques”

Cheng, column 6, lines 40-50.

[T]he user logs in 201 to the service provider computer 102 with the client application 104 . . . providing a user ID [and] a password . . . If the user is not registered, then the service provider computer 102 in conjunction with inputs by the user, registers 202 the new user of the system. . . . The user may select check box 309 to indicate that they want to be notified by email when new software updates are available for software products installed on their computer. When the registration process 202 is completed, the service provider computer 102 returns a unique registration number to the user. This number may be stored on the client computer 101 and used during subsequent logins to identify the user to the service provider computer 102

Cheng, column 7 lines 11-39; *see also* Figures 2 and 3.

The user profile database 711 . . . stores information descriptive of each user . . . [including information] for use by the security 701, communications 703, and payment 705 modules. In a user table 1400, each user is identified by user ID 1401, name 1403, email address 1405, the start date 1407 of their subscription to the service, the end or termination date 1409 of the subscription, credit card information 1411 such as number, issuer and expiration date, a user selected password 1413, and a public key 1415 or other authentication token. As illustrated in FIG. 3, the user has the option 309 of requesting notification by email of such software updates. The user table 1400 thus also includes a flag 1416 indicating whether the user so desires to be notified by email. The user table 1400 is keyed by the user ID 1401 to a notification table 1417 that associates the user with selected product names 1419 and their current version 1421. When a software vendor or the service provider updates the update database 709 with information for a new software update, the notification table 1417 may be scanned to identify those users by

user ID 1401 to notify about the update. The email flag 1416 for a user is checked, and if true, the user's email address 1405 is obtained from the user table 1400 and the user notified by email with information identifying the new software update
Cheng, column 20, lines 33-60.

As evidenced by the above excerpts, Cheng does not disclose any one item “maintaining, on a server for each of a plurality of computing systems, data specifying which resources are authorized for the computing system.” Rather, Cheng discloses registering a user via a client computer for a service including storing information about the user in a database at a service provider computer. Cheng further discloses that this user information may be used by the payment module at the service provider computer, and that a payment mechanism is provided by software vendor computers. Cheng provides evidence of users being authorized by authenticating, not “which resources are authorized for the computing system” as is claimed.

(b) Rejection of Dependent Claims

THIRD GROUND OF REJECTION (addressed in detail, below): The Interview Summary of July 10, 2007 (pertaining to the interview of June 27, 2007) states that claims 1, 2, 27, and 28 were discussed; it further states:

Applicant's Representative [and the Examiner] discussed the proposed claim amendments. Examiner indicated to [Applicant's Representative] that the specification does not provide enough support for claim 1 [incorporating subject matter of claims 2 and 28] as proposed by Applicant.

Applicant's Representative suggested using the features disclosed in specification on page 11 or in original claim 27 to amend the claims to recite limitations with respect to a new computing system. Examiner notified Applicant's Representative that the implementation discussed on page 11 merely refers to automatic installation and configuration and no assessment nor determination of authorized resources is made to provide the resources to the new computing system.

The Examiner's position appears inconsistent with the evidence of record. For example, the Specification, states:

It will be appreciated, from the description to follow, that the unique identifier need not come from the computing system to be configured/reconfigured. Rather, the unique identifier may well come from another computing system or communication system element. In such an implementation, configuration agent 14 may well configure a first computing system (e.g., installing an operating system, applications and settings on a new computer system), while updating the resources of the computing and/or communication system element used to issue the unique identifier initiating the configuration process.

See Application, Page 5, line 18-25.

According to one implementation of the present invention, a new computing system (e.g., computing system 18A) is provided to an end-user without any operating system, applications and/or configuration information pre-installed on the computer system. Rather, computing system 18A is provided with a fundamental level of functionality that minimally includes a BIOS 306 and the configuration interface 308. According to one implementation, configuration interface 308 transmits a unique identifier to the configuration agent 14 at server 12 to facilitate the automatic configuration of a host computing device. In addition, configuration interface 308 is used by configuration agent 14 to initially receive one or more of an operating system image, application(s), and

configuration/personalization settings as configuration agent installs and configures the host computer with such resources.

According to one implementation, configuration interface 308 includes a user interface, which minimally enables an end-user to locate/specify a server 12 incorporating a configuration agent 14. In certain implementations, configuration interface 308 includes one or more elements of configuration agent 14 such as, for example, identification and characterization function (to be discussed more fully below with reference to configuration agent 14), to locally assess the hardware and/or software resources of computing system 18, and provide the assessment to configuration agent 14. In such an implementation, configuration agent 14 utilizes the assessment to identify additional resources and/or configuration information, and downloads the identified resources to configuration interface 308 for installation on the host computing system to enhance the operational capability of the host computing system.

See Application, Page 11, line 9 to Page 12, line 7.

In addition, the Interview Summary of February 23, 2006 (pertaining to the interview of January 25, 2006) states:

“... it appears to the Examiner that the initial implementation of ‘when the user receives a new computing system without any applications pre-installed . . . a configuration agent initially installs one or more applications’ as described on page 11, second paragraph . . .”

While Appellant has claimed subject matter as suggested by the Examiner, the Examiner maintains an interpretation of the claim language to fit the reference cited rather than in view of the Specification.

THIRD GROUND OF REJECTION:

Claim 2:

Cheng: Evidence of “without the authorized resources being preinstalled”

While the Office's grounds for rejecting the independent claims do not specifically rely on evidence of “without the authorized resources being preinstalled” in Cheng, the Office's rejection of dependent claim 2 does rely on such evidence. Appellant asserts that Cheng provides insufficient evidence to support disclosure of the recited “without the authorized resources being preinstalled” of claim 2 (as discussed in detail below).

For ease of reference, claim 2 recites “*A method according to claim 1, wherein the computing system is provided to the user without the authorized resource being preinstalled.*”

While Cheng discloses products which are not preinstalled, Cheng's users must express interest in receiving notification of such products, and Cheng does not disclose such products being *authorized resources* for the computer system at the time the computing system is provided to the user. Cheng states:

Each client computer 101 is operated by an end user, and typically has a number of software products installed thereon, such as applications, drivers, utilities and the like. In accordance with the present invention, the client computers 101 includes a client application 104 that communicates with the service provider computer 102 to obtain software updates of software products installed on the client computer 101. The software architecture of a client computer 101 and client application 104 is further described below with respect to FIG. 7.

Cheng, column 6, lines 20-30.

When a software vendor or the service provider updates the update database 709 with information for a new software update, the notification table 1417 may be scanned to identify those users by user ID 1401 to notify about the update. The email flag 1416 for a user is checked, and if true, the user's email address 1405 is obtained from the user table 1400 and the user notified by email with information identifying the new software update.

Cheng, column 20, lines 52-60.

The user profile database 711 maintains a profile for each user containing information about which products the user has shown an interest, for example by requesting notification about software updates for specific products, or about new software products. This information is then used to deliver notifications about new updates available for these products to the user, for example by email, or other electronic communications mechanisms. This optional feature of the service provider computer 102 further enhances the value of the service to the user, ensuring timely notification of the availability of software updates and new software products.

In this regard, one alternate embodiment of the present invention is the use of email to notify users about new software update information, and new software products for which the user has expressed an interest. Specifically, when a new software update or software product is available, the service provider computer 102 sends an email to those users who have requested notification by email. The email contains information about the software update, and may include the record from the update table 807 about the software update, including the URL data 823 used to access the software update files. The client application 104 would then read the update information, and verify that the software update is indeed applicable to the client computer 101, and that the client computer 101 satisfies any conditions for installation. If the software updates are approved by the user, the client application 104 downloads the software update, verifies

its integrity, and installs the software update directly, without having to login 201 to the service provider computer 102, and analyze 204 the software products installed on the client computer 101. In the case of notifications about new software products in which the user had expressed interest, the client application 104 would verify that the user is still interested in the software product and proceed to purchase, download and install it.

As a further enhancement of the e-mail notification embodiment, the email sent by the service provider computer 102 includes a specification of conditions a client computer 101 must satisfy for the software update or software product to be installed. This information is essentially the same as that used by the client application 104 to determine the relevant software updates for the client computer 101. For example, this information includes, for a software update, the older versions of the software product to which it is applicable. This additional information in the email notification is used by the client application 104, for example, to ensure that the software update is used only once by the user, and can be repeatedly applied.

Cheng, columns 19, lines 50-67 and column 20 lines 1-32.

As evidenced by the foregoing excerpts, even the products which are not preinstalled in Cheng are not *authorized*. After a user has expressed interest, the system of Cheng notifies a user via email about the product availability and a client application verifies that the user is still interested in the product.

Appellant submits that the products not preinstalled that are disclosed by Cheng are not authorized resources at the time the computing system is provided to the user. Based on this evidence, Appellant maintains that Cheng does not disclose "the computing system is provided to the user without the authorized resources being preinstalled."

Claim 28:

Cheng: Evidence of “the computing system is an unconfigured computing system”

While the Office's grounds for rejecting the independent claims do not specifically rely on evidence of “the computing system is an unconfigured computing system” in Cheng, the Office's rejection of dependent claim 28 does rely on such evidence. Appellant asserts that Cheng provides insufficient evidence to support disclosure of the recited “the computing system is an unconfigured computing system” of claim 28 (as discussed in detail below).

For ease of reference, claim 28 recites “*A computing system according to claim 27, wherein the computing system is an unconfigured computing system.*”

While Cheng discloses configuring computer systems related to obtaining and installing software updates, configuration in Cheng is directed to overcoming user reluctance to disclose configuration information necessary to determine which updates would be appropriate for the user's computer system, i.e., what hardware and software components makes up the particular system and what updates apply to that configuration. Cheng does not disclose such computing systems being *unconfigured computing systems*. Cheng states:

[M]any users have concerns about their privacy, and are often resistant to revealing complete information about their software configurations to one or more vendors. However, even for a single vendor, information about which of the vendor's products are installed on a user's computer system, and system

configuration information is necessary for determining which updates are applicable to the user's computer system. For example, a certain software update to an accounting program from vendor A might be applicable if the user has a printer from vendor B, and a different software update is applicable if the printer comes from vendor C. The user might not want to let each vendor know about all the components on their system, but this configuration information is necessary to ensure the correct software updated is installed. Still, users are resistant to the prospect of a single vendor storing information profiling the software components that reside on their computer systems.

Cheng, column 2, lines 24-40, *Background*.

[Cheng's] system allows numerous users to periodically and automatically update the software products on their computers from diverse software vendors through a single, update mechanism. The users need not invest the time and energy to identify currently available updates, nor engage in the potentially difficult process of manually (even electronically) obtaining and installing the software updates, and properly configuring their computer systems. Rather, all of the relevant information about the currently available updates is maintained for subscribing users in the service provider's database.

Cheng, column 4, lines 12-23, *Summary*.

The update database of software update information is preferably maintained by a supplier of the client application, who also maintains the service of providing software updates. This database is created by engaging with numerous software vendors to provide information for software updates of their products to the service provider. Preferably [sic], the software vendors provide the service provider information describing the software update, including version information, file formats, configuration information, and network location.

Alternatively, software updates may be identified by systematically and periodically searching the Internet to identify the software vendors who are providing updates to their software products. These updates are then downloaded from the software vendor's Internet sites, and one or more network

locations (URL) are identified for obtaining the download. The downloaded software update is then installed on one or more target computers as typically configured by end users. The format of the software update is determined, along with the network location of the software update, and other descriptive information, such specific configuration actions to be taken during installation of the software update, useful descriptive text, and the like.

Cheng, column 4, lines 50-67 and column 5, lines 1-6, *Summary*.

Once the download and applicable payment are complete, the software update is physically installed on the client computer 101. Each software [sic] update is associated with information that describes the particulars for the installation, such as configuration, decompression or other information. The installation is performed in conformance with such information.

In the preferred embodiment, the client application 104 executes 211 an install monitor prior to actually installing the software update. The install monitor, . . . records the changes made to the client computer 101 as a result of the installation of the software update. This information is archived by the install monitor and allows the user to "undo" or remove any number of installations, and restore the client computer 101 to its state prior to each such installation. Accordingly, the client application 104 performs 212 the installation, executing any necessary decompression, installation, or setup applications necessary to install the software update. During the installation process 212 the install monitor records 213 any changes made to the system configuration, including changes to various configuration files, additions or deletions of files, and additions or deletions of directories. The changes may be recorded in a variety of manners, such as building descriptions of the modifications of the files, or alternatively, storing copies of files prior to their alteration or deletion. Once the installation is complete, the install monitor archives 214 the changes. This process 208 is repeated for each software update to be installed.

Cheng, column 8, lines 55-67 and column 9, lines 1-16.

At some subsequent point, the user may decide to undo a previous installation, for example, due to dissatisfaction with the software product. The user may use a recovery feature of the client application 104 to undo 216 the installation. A sample user interface 600 for the recovery function is illustrated in FIG. 6. The user interface 600 includes a field 601 indicating the previous update to be removed as selected by the user, along with an information window 603 describing the software update. The user confirms the removal of the software update by selecting the undo button 605, or may cancel with cancel button 607. The recovery function deletes the files installed for the software update, and using the archived information created by the install monitor during the installation of the product, restores the client computer system 101 to its configuration immediately before the installation of the product. This process 216 includes deleting files and directories that were added, restoring files and directories that were deleted, and restoring files that were otherwise changed. In one preferred embodiment, the recovery function is able to undo any installation in a given series of installations, accounting for changes to the configuration of the client computer 101 after a particular installation.

Cheng, column 9, lines 28-49.

Finally, the update database 709 may also store information describing an installation process for installing a software update. This information may include particular configuration, file format, or other data useful to performing the installation of the software update [on] the client computer 101. This information, if present, may be provided to the client computer 101 to use during the installation of the software update.

Cheng, column 10, lines 55-62.

The product locator table 803 associates individual scan.sub.--strings 813 with a product name 815, instructions 816 for determining a version number or release number, and one or more constraints 814. The constraint is a rule that uniquely identifies the product given contextual information for the product where there are two entries having identical scan.sub.--strings. Constraints include specific directories that include the

product, additional entries in the system configuration file, the Registry or the like. If the specified information in these various locations matches the constraint values, then the product name associated with the constraint is the correct product name for the scan.sub.-- string. In one embodiment, the constraint 814 is an executable procedure that retrieves information in these various locations, and determines from this information whether the product name is a match with the scan.sub.-- string, according to whether the specified details of the constraint are found in the client computer 101.

Cheng, column 11, lines 38-63.

The client computer 101 includes some number of configuration files 915, such as the Windows95 Registry, the system.ini, config.sys and other files.

The client computer 101 further has installed thereon software products in the form of applications 912, operating system utilities 913, and device drivers 914, and the like. These various software products are among those that will be updated by the service provider computer 102.

Cheng, column 13, lines 21-28.

Once the tables are synchronized, the system analyzer 907 can operate locally, for improved efficiency. The system analyzer 907 traverses the entire method table 801, and invokes 1003 each scan method 812 to search the Registry and configuration files 915 of the client computer 101. Each scan method 811 outputs a scan.sub.-- string, as described, specifying some software product installed on the client computer 101.

Cheng, column 14, lines 8-15.

The software update is then installed 212 by the client application 104 as described, using the format information 825 to determine the particular installation process, and the script 826 to control any custom installation or configuration information.

FIG. 11 illustrates a flowchart of the operation of the install monitor 910. The install monitor 910 receives operating system calls and messages from the client application 104. On

trapping 1101 an operating system call, the install monitor 910 determines 1103 the type of call. There are three types of calls of interest: calls 1105 that delete a file or directory, calls 1107 that change an existing file by writing to it, and calls 1109 to add new a file or directory. When a file or directory is to be deleted, the install monitor 910 first makes 1113 a copy of the existing file or directory to a private area of the client computer's 101 hard disk or other storage device. The install monitor 910 then lets the operating system 917 delete the file or directory, and waits for the next call. When a file is to be changed 1107, the install monitor 910 determines 1115 whether this is the first write to the file. If so, then again, the install monitor 910 copies 1119 the file to the private area. If the file has been already changed during the installation, there is no need to copy it again. These copy operations 1113, 1119 preserve the configuration of the client computer 101 prior to the installation. Finally, if a new file or directory is to be added 1109, the install monitor 910 stores 1117 the pathname of the new file or directory. This allows the new file or directory to be later deleted during an undo of the installation. For all other types 1111 of operating system calls, the install monitor 910 passes them through without action.

The install monitor 910 waits for installation process 212 to complete, preferably indicated by a message from the client application 104. At this point the complete prior configuration of the client computer 101 is known from the copied files and pathname information. These files and information are compressed 1121 into an archive file 909 and saved on the client computer 101, along with information identifying the software product installation to which it belongs. This identifying information allows the recovery module 908 to retrieve the archived information and restore the configuration of the client computer 101.

Cheng, column 15, lines 32-36 and column 15, lines 47-67 to column 16, lines 1-15.

As evidenced by the foregoing excerpts, Cheng does not address an *unconfigured computing system* as claimed. Cheng addresses configuration of a computer in terms of

configuration files and the analysis in Cheng takes place on the client computer by a client application. Thus, Cheng cannot anticipate an unconfigured computing system as is claimed.


Appellant submits that the configuration disclosed by Cheng is not an unconfigured computing system, for example when a new computing system is provided to the user. Based on this evidence, Appellant maintains that Cheng does not disclose "*A computing system . . . wherein the computing system is an unconfigured computing system.*"

CONCLUSION

The Applicant respectfully considers this application to be in condition for allowance. Appellant submits, as evidenced by the above discussion, Claims 1-36 satisfy the requirements of 35 U.S.C. § 102(e) and therefore are not anticipated by Cheng. Therefore, Appellant respectfully requests that the Board overturn the final rejection and that the Examiner pass this application to allowance.

Dated this 27th day of December, 2007.

Respectfully submitted,


BEATRICE L. KOEMPEL-THOMAS
Attorney for Appellant/Applicant
Registration No. 58,213

LEE & HAYES PLLC
421 W. Riverside Avenue, Suite 500
Spokane, WA 99201
Telephone: (509) 324-9256 (Ext. 249)
Facsimile: (509) 323-8979

VIII. APPENDIX: CLAIMS ON APPEAL

1. **(Previously Presented)** A method comprising:
 - maintaining, on a server for each of a plurality of computing systems, data specifying which resources are authorized for the computing system;
 - receiving by a configuration agent at the server an identifier associated with a computing system and/or computing system user ,the configuration agent:
 - obtaining, from the server, data that specifies authorized resources corresponding to the received identifier;
 - interrogating the computing system to produce an assessment indicating existing hardware and/or software resources available on the computing system;
 - comparing the authorized resources with the assessment to identify one or more resources authorized but not installed on the computing system; and
 - automatically modifying the computing system resources by installing the one or more identified resources.
2. **(Previously Presented)** A method according to claim 1, wherein the computing system is provided to the user without the authorized resources being preinstalled.

3. **(Original)** A method according to claim 1, wherein the identifier associated with a computing system and/or computing system user is received from the computing system.

4. **(Previously Presented)** A method according to claim 1, wherein the identifier associated with the computing system and/or computing system user is received from the computing system and/or a communications device associated with the computing system user, wherein the communications device is not coupled directly to the computing system.

5. **(Previously Presented)** A method according to claim 4, further comprising:
 automatically modifying system resources of the communications device based, at least in part, on the assessment of the computing system resources.

6. **(Previously Presented)** A method according to claim 1, further comprising :
 selectively updating certain of the computing system resources based, at least in part, on the comparison of the assessed computing system resources against authorized and available computing system resources.

7. **(Original)** A method according to claim 1, wherein the computing system is a communications device, the method further comprising:

assessing communications device resources;

comparing the assessed communications device resources against authorized and available communications device resources; and

selectively installing, configuring and/or updating one or more communications device resources based, at least in part, on the assessed communications resources.

8. **(Previously Presented)** A method according to claim 1, wherein the identifier is received from the computing system and/or a communications device associated with the computing system user remote from the computing system, the method further comprising:

automatically modifying communications device resources based, at least in part, on an assessment of the communications device resources.

9. **(Original)** A method according to claim 8, wherein the identifier is one or more of a telephone number associated with the user, an electronic serial number (ESN) of the communications device associated with the user, an electronic identifier associated with the computing system, and/or a serial number associated with one or more hardware and/or software resources of the computing system.

10. (Original) A method according to claim 1, wherein the identifier is one or more of a telephone number associated with the user, an electronic serial number (ESN) of a communications device associated with the user, an electronic identifier associated with the computing system, and/or a serial number associated with one or more hardware and/or software resources of the computing system.

11. (Original) A storage medium comprising a plurality of executable instructions which, when executed, implement a method according to claim 1.

12. (Original) A server comprising:
a storage device having stored therein a plurality of executable instructions; and
a control unit, coupled to the storage device, to execute at least a subset of the plurality of executable instructions to implement a method according to claim 1.

13. (Previously Presented) A server comprising:

a storage device to maintain a profile of personal resources specifying, for each of a plurality of computing systems, which resources are authorized for the computing system; and

a configuration agent, coupled to the storage device, to:

receive an identifier associated with a computing system and/or computing system user;

generate an assessment of the current resources of the computing system;

identify, by comparing the assessment with the authorized resources, one or more of the authorized resources which are missing from a computing system ; and

automatically configure resources of the computing system to include the identified resources.

14. (Previously Presented) A server according to claim 13, wherein an assessment of the computing system resources comprises an assessment of at least one of an operating system, configuration settings, personalization settings, Internet settings or application settings on the computing system.

15. (Previously Presented) A server according to claim 13, wherein the profile includes a list of identifiers associated with authorized users and the configuration agent

accesses a user profile on the storage device based, at least in part, on the identifier.

16. (Previously Presented) A server according to claim 13, wherein the configuration agent receives the identifier from the computing system and/or a communications device remote from the computing system associated with the computing system user.

17. (Previously Presented) A server according to claim 16, wherein the configuration agent further automatically modifies communications device resources based, at least in part, on an assessment of communications device resources.

18. (Previously Presented) A server according to claim 13, wherein the configuration agent is further configured to update the computing system resources.

19. (Original) A server according to claim 13, wherein the identifier is one or more of a telephone number associated with the user, an electronic serial number (ESN) of a communications device associated with the user, an electronic identifier associated with the computing system, a serial number associated with one or more hardware and/or software resources of the computing system.

20. (Previously Presented) A server according to claim 13, wherein the storage device includes a plurality of executable instructions, the server further comprising:

a controller, coupled to the storage device, to execute at least a subset of the plurality of executable instructions to implement an instance of the configuration agent.

21. (Previously Presented) A storage medium comprising a plurality of executable instructions including at least a subset of which that, when executed, implement a configuration agent at a server to:

maintain, for each of a plurality of computing systems, data specifying authorized resources for the computing system;

conduct an assessment of computing system resources upon receipt of an identifier associated with the computing system and/or computing system;

identify, by comparing the assessment with corresponding data specifying authorized resources, one or more of the authorized resources which are missing from the computing system; and

automatically download and install on the computing system the missing authorized resources.

22. (Previously Presented) A storage medium according to claim 21, wherein the configuration agent is further configured to update computing system resources.

23. (Original) A storage medium according to claim 21, wherein the configuration agent interrogates the computing system upon receipt of the identifier to assess computing system resources.

24. (Previously Presented) A storage medium according to claim 23, wherein the configuration agent modifies the computing system resources to include available and authorized resources based, at least in part, on the assessment of the computing system resources.

25. (Original) A storage medium according to claim 21, wherein the computing system is a communications device.

26. (Previously Presented) A storage medium according to claim 21, wherein the identifier is received from a communications device remote from the computing system, and wherein the configuration agent automatically modifies computing system resources and communications device resources based, at least in part, on assessment of system resources of the computing system and communications device.

27. (Previously Presented) A computing system comprising:

a storage device having stored thereon a plurality of executable instructions;
a network interface, communicatively coupling the computing system to a network; and

a controller, coupled to the storage device and the network interface, to execute at least a subset of the plurality of executable instructions to make an assessment of current hardware and/or software resources of the computing system, and to implement a basic input/output system (BIOS) to issue a configuration request to the network via the network interface, the configuration request based on the assessment and including an identifier associated with the computing system, wherein the configuration request is configured to cause a recipient of the request to:

reference the identifier to access corresponding data specifying authorized resources associated by the identifier with the computing system;

compare the assessment to the authorized resources to determine one or more of the authorized resources missing from the computing system; and

provide the missing authorized resources to the computing system via the network.

28. (Original) A computing system according to claim 27, wherein the computing system is an unconfigured computing system.

29. **(Previously Presented)** A computing system according to claim 27, wherein the controller receives one or more commands to receive and install computing system resources from network devices remote from the computing system via the network interface in response to the configuration request.

30. **(Original)** A computing system according to claim 27, wherein the identifier is associated with the computing system and/or computing system user.

31. **(Original)** A computing system according to claim 27, wherein the computing system is a communications device.

32. (Previously Presented) A method comprising:

issuing a configuration request from a computing system, wherein the configuration request includes an identifier associated with the computing system and/or computing system user and is configured to cause a recipient of the request to:

generate an assessment of the current computing system resources of the computing system;

reference the identifier to access data specifying authorized computing system resources associated by the identifier with the computing system; and

compare the assessment to the authorized computing system resources to determine one or more of the authorized computing system resources missing from the computing system; and

receiving a response to the configuration request at the computing system, the response including the one or more computing system resources missing from the computing system, wherein the one or more computing system resources are automatically installed and configured on the computing.

33. (Original) A method according to claim 32, wherein the one or more computing system resources are automatically installed and configured in response to installation and configuration commands received from a remote computing system.

- 34. (Previously Presented)** A method according to claim 32, wherein the computing system is a communications device.
- 35. (Original)** A method according to claim 34, wherein the one or more system resources enable the communications device to communicate over an additional communications medium.
- 36. (Previously Presented)** A method according to claim 32, wherein the configuration request is issued from a communications device remote from the computing system associated with the computing system user, the method further comprising:
- receiving a response to the configuration request at the communications device including one or more computing system resources, wherein the one or more computing system resources are automatically installed and configured on the computing system.

IX. APPENDIX: EVIDENCE

None.

X. APPENDIX: RELATED PROCEEDINGS

None.